

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-046383

(43)Date of publication of application : 26.02.1993

(51)Int.Cl.

G06F 9/34

(21)Application number : 03-209112

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 21.08.1991

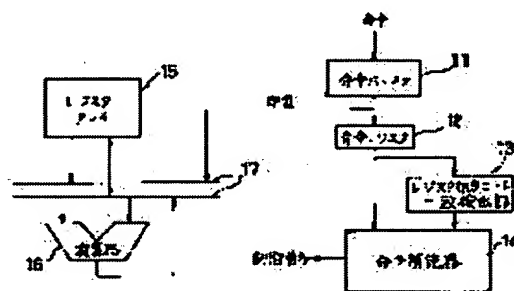
(72)Inventor : MATSUZAKI TOSHIMICHI
DEGUCHI MASASHI

(54) DATA PROCESSOR

(57)Abstract:

PURPOSE: To provide the data processor which can suppress a program size small by constituting the processor so that many kinds of operations can be executed by short instruction word length.

CONSTITUTION: This processor is constituted so that a register designation code coincidence detecting part 13 detects whether two register designation codes contained in an instruction held by an instruction register 12 coincide with each other or not, an instruction decoder 14 executes decoding of an instruction, and in accordance with a results of the detection, in the case two register designation codes are different from each other, a computing element 16 is allowed to calculate to calculate data of two registers designated by each register designation code, and on the other hand, in the case the designation codes are equal to each other, the data of the designated register and an immediate value outputted from an instruction buffer 11 are calculated. Accordingly, since two kinds of processings can be executed with respect to the same operation code, many kinds of operations can be executed by short instruction word length.



LEGAL STATUS

[Date of request for examination]

07.05.1997

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

2970821

[Date of registration]

27.08.1999

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-46383

(43)公開日 平成5年(1993)2月26日

(51)Int.Cl.⁵

G 0 6 F 9/34

識別記号

3 3 0

庁内整理番号

9189-5B

F I

技術表示箇所

審査請求 未請求 請求項の数2(全 8 頁)

(21)出願番号 特願平3-209112

(22)出願日 平成3年(1991)8月21日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 松崎 敏道

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 出口 雅士

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74)代理人 弁理士 中島 司朗

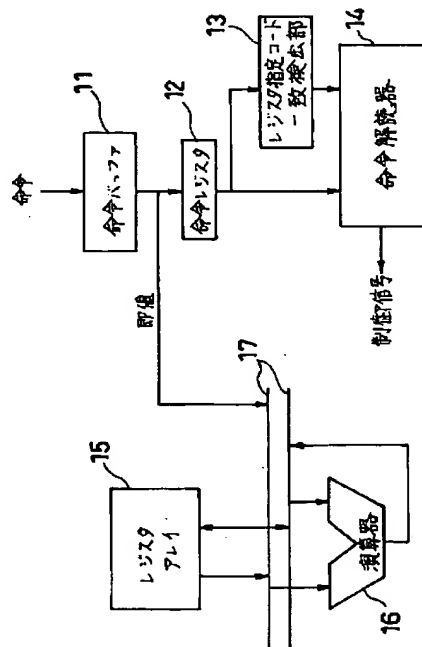
(54)【発明の名称】 データ処理装置

(57)【要約】

【目的】 短い命令語長で多くの種類の演算を実行させ得るようにして、プログラムサイズを小さく抑えることができるデータ処理装置を提供する。

【構成】 レジスタ指定コード一致検出部13は、命令レジスタ12に保持された命令に含まれる2つのレジスタ指定コードの一致の有無を検出し、命令解読器14は命令の解読を行い、上記検出結果に応じて、2つのレジスタ指定コードが互いに異なる場合には、演算器16に各レジスタ指定コードによって指定される2つのレジスタのデータを演算させる一方、互いに等しい場合には、指定されるレジスタのデータと命令バッファ11から出力される即値とを演算させるように構成されている。

【効果】 同一のオペレーションコードに対して2通りの処理を行わせることができるので、短い命令語長で多くの種類の演算を実行させることができる。



【特許請求の範囲】

【請求項1】 オペレーションコードと、2つのレジスタ指定コードとから成るインストラクションに基づいてデータ処理を行うように構成されたデータ処理装置であって、

前記2つのレジスタ指定コードが互いに異なる場合と同一である場合とで、異なる処理を行わせるインストラクション実行制御部を備えたことを特徴とするデータ処理装置。

【請求項2】 データを保持するm本のデータレジスタと、アドレスを保持するn本のアドレスレジスタとを備え、前記オペレーションコードは、所定の最小長さのコードで、メモリとデータレジスタ間のデータ転送およびデータレジスタ間の演算を示すとともに、前記レジスタ指定コードは、log m、およびlog nの長さのコードで、処理対象となるレジスタを示すように設定されていることを特徴とする請求項1のデータ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、CPUやマイクロプロセッサなどと称されるデータ処理装置に関するものである。

【0002】

【従来の技術】従来のデータ処理装置には、演算命令の演算対象として、データを格納するメモリだけを用いるように構成されたものがある。この種のデータ処理装置では、演算内容を示すオペレーションコード、および演算対象となるメモリを示すための2つのオペランドアドレス（すなわち、例えばソースオペランドアドレスとデスティネーションオペランドアドレス）から成るインストラクションに基づいて、所定の演算が行われるようになっている。

【0003】また、アキュムレータと称されるレジスタ、およびその他のレジスタを備え、主な演算の対象として上記アキュムレータが含まれるように構成されたデータ処理装置も用いられている。この種のデータ処理装置では、演算内容を示すオペレーションコード、およびアキュムレータ以外の演算対象となるレジスタを示す1つのレジスタ指定コードから成るインストラクションに基づいて、所定の演算が行われるようになっている。

【0004】また、任意の組み合わせで演算対象となり得る汎用レジスタを備え、演算内容を示すオペレーションコード、および演算対象となるレジスタを示す2つのレジスタ指定コードから成るインストラクションに基づいて、所定の演算を行うように構成されたものも用いられている。

【0005】

【発明が解決しようとする課題】しかしながら、上記メ

モリだけを演算対象として用いるデータ処理装置では、インストラクションが2つのオペランドアドレスの設定されるオペランドフィールドを必要とするため、命令語長が長く、したがってプログラムサイズが大きくなりがちである。

【0006】また、アキュムレータを備えたデータ処理装置では、インストラクションが、オペランドアドレスよりもビット長の短い1つのレジスタ指定コードが設定されるレジスタフィールドしか必要としないので、命令語長は短いものの、主な演算はアキュムレータを対象として行われるために、アキュムレータと他のレジスタまたはメモリとの間の転送を頻繁に行う必要があり、命令数が多くなってやはりプログラムサイズが大きくなりがちであるうえ、処理速度も低下しがちである。

【0007】一方、汎用レジスタを備えたデータ処理装置では、汎用レジスタを任意の組み合わせで演算対象とすることができるので命令数を比較的小さく抑えることができ、また、オペランドアドレスよりもビット長の短いレジスタ指定コードを用いるため、命令語長も比較的短い。しかしながら、このようなデータ処理装置でも、以下に示すように、短い命令語長で、しかも多くの種類の演算を行えるようにして命令数を少なく抑え得るよう構成することは困難であるという問題点を有している。

【0008】すなわち、例えば8本の汎用レジスタが備えられている場合、1つのレジスタを特定するためのレジスタ指定コードのビット長は3ビットとなり、インストラクションは合計6ビットのレジスタフィールドを必要とする。この場合、例えば命令語長を8ビットにすることは、インストラクションが最大でも4種類しか用いられないことになるので実用的ではなく、実際上は命令語長を16ビットにする必要がある。したがって、命令語長を短くするとともに、多くの種類の演算を行えるようにすることは困難である。

【0009】本発明は上記の点に鑑み、短い命令語長で多くの種類の演算を実行させ得るようにして、プログラムサイズを小さく抑えることができるデータ処理装置の提供を目的としている。

【0010】

【課題を解決するための手段】上記目的を達成するため、本発明は、オペレーションコードと、2つのレジスタ指定コードとから成るインストラクションに基づいてデータ処理を行うように構成されたデータ処理装置であって、前記2つのレジスタ指定コードが互いに異なる場合と同一である場合とで、異なる処理を行わせるインストラクション実行制御部を備えたことを特徴としている。

【0011】

【作用】上記の構成により、インストラクション実行制御部の制御によって、2つのレジスタ指定コードが互い

に異なる場合と同一である場合とで異なる処理が行われる。

【0012】

【実施例】以下、本発明の一実施例を図1ないし図5に基づいて説明する。図1はデータ処理装置の要部の構成を示すブロック図である。同図において、命令バッファ11は、図示しないメモリから読み出される命令（インストラクション）を一時的に蓄えるものである。

【0013】命令レジスタ12は、上記命令バッファ11に蓄えられた命令を、さらに命令の解釈、または実行が完了するまでの間保持するものである。レジスタ指定コード一致検出部13は、命令に2つのレジスタ指定コードが含まれている場合に、両者が等しいかどうかを検出するものである。命令解読器14は、上記命令レジスタ12に保持された命令の解釈を行って、データ処理装置各部の動作を制御する制御信号を出力するものである。ここで、命令解読器14は、命令に含まれるオペレーションコードが所定のオペレーションコードである場合には、後述するように上記レジスタ指定コード一致検出部の検出結果に応じて、レジスタ指定コードが一致する場合と一致しない場合とで異なる制御を行うようになっている。

【0014】レジスタアレイ15は、図2に示すように、それぞれ16ビット長のデータレジスタD0～D3から成るデータレジスタ組、アドレスレジスタA0～A3から成るアドレスレジスタ組、実行中の命令の格納アドレスを保持するプログラムカウンタPC、およびデータ処理装置の状態を示す状態レジスタPSW（processor status word）が設けられて構成されている。

【0015】上記データレジスタD0～D3、およびアドレスレジスタA0～A3は、それぞれデータまたはアドレスを保持し、後述する演算器16の演算に用いられるようになっている。より具体的には、例えば、データレジスタD0～D3は主として演算における作業用として用いられる一方、アドレスレジスタA0～A3は、それぞれ、ソースオペランド用、ディスティネーションオペランド用、フレームポインタ用、またはスタックポインタ用として用いられる。

【0016】各データレジスタD0～D3、およびアドレスレジスタA0～A3は、命令コード中で、00～11（2進数表記）のレジスタ指定コードによって指定されるようになっている。また、上記状態レジスタPSWは、より詳しくは例えば図3に示すように、デバッグモードで動作中にプログラムのシングルステップ実行を指定するトレースフラグTと、16レベルの割り込みマスクを指定する割り込みマスクフラグIM3～IM0と、演算の結果を反映するオーバフローフラグV、キャリフラグC、ネガティブフラグN、およびゼロフラグZから成っている。

【0017】演算器16は、レジスタアレイ15から出

力されるデータやアドレス、命令バッファ11から出力される即値、および図示しないメモリから読み出されたデータ等が内部データバス17を介して入力され、命令解読器14からの制御信号に従って算術演算や、論理演算、転送等の所定の演算を行うものである。演算器16の演算結果は、内部データバス17を介してレジスタアレイ15やメモリに送られ、命令に応じた所定のレジスタ等に保持されるようになっている。

【0018】なお、データ処理装置には、上記の他にも図示しないデータ処理装置外部のデータバスに接続されるデータバスインタフェースや、アドレスバスに接続されるアドレスバスインタフェースなどが設けられているが、本発明には直接関係しないので、その説明を省略する。次に、上記データ処理装置に適用される命令コードの構造、およびオペレーションコードの割り当ての例を図4および図5に基づいて説明する。ここで、命令体系としては、装置の構成を簡素化し得るとともに処理の高速化が容易なロードストアの命令体系、すなわち各レジスタとメモリとの間ではデータ転送だけが可能な命令体系が適用される場合の例を示す。

【0019】図4は命令コードの構造の例を示すもので、(a)は、4ビットのオペレーションコードと、それぞれ2ビットの2つのレジスタ指定コード（D、およびD'、またはD、およびA、）とから成る構造を示す。

(b)は、(a)と同様のオペレーションコードおよびレジスタ指定コードに加えて、8ビットの即値（imm8）、アドレスのディスプレースメント（d8）、または8ビットの絶対アドレス（abs8）から成る構造を示す。

【0020】(c)は、6ビットのオペレーションコードと、1つの2ビットのレジスタ指定コード（D、）とから成る構造を示す。

(d)は、8ビットのオペレーションコード、および8ビット×1～3の拡張命令や即値、アドレス指定のディスプレースメント等の付加部から成る構造を示す。

【0021】図5は命令コードの割り当ての例を示すもので、左から順に、命令コード、命令コードの構造（図4に示した(a)～(d)の何れであるか）、命令タイプ、および演算の内容が対応付けて表わされている。ここで図5において、「***」および「**」は、その3ビットまたは2ビットの値に応じて、演算内容の欄に示す何れかの演算が行われることを示すものである。

【0022】以下、各命令コードについて命令タイプごとに説明する。命令タイプ[R-R]は、2つのデータレジスタ間の2項演算命令で、命令コード「0***D、D、'」（ただし、D、≠D、'）で表わされ、3ビットの「***」の値に応じて、(1)データレジスタ指定コードD、'で示されるデータレジスタ（以下、単にデータレジスタD、'等と称する。）から、データ

5

レジスタD、へのデータ転送、(2)両データレジスタに保持されているデータのキャリ無し加算、(3)同、ボロー無し減算、(4)同、比較、(5)同、キャリ付き加算、(6)同、ボロー付き減算、(7)同、論理積、(8)同、論理和

の8種類の演算命令であることを示している。なお、上記各演算についての演算結果は、データレジスタD、に格納される。

【0023】命令タイプ[R-imm8]は、上記命令タイプ[R-R]の命令とは、2つのレジスタ指定コードが互いに等しい(D.=D.)ことによって区別され、命令タイプ[R-R]と同様の2項演算が、データレジスタD、のデータと、命令コードの「0***D、D。」に続く8ビットの即値(imm8)とについて行われることを示す。ここで、8ビットの即値が演算に用いられる場合でも、演算器16には16ビットのデータが入力されるが、その上位8ビットは演算の種類に応じて0拡張(例えば論理演算の場合)や、符号拡張(例えば算術演算の場合)などが行われるようになっている。

【0024】命令タイプ[LD]、[ST]は、それぞれ、メモリからデータレジスタD、へのデータ転送命令(load)、またはその逆のデータ転送命令(store)を示し、メモリにおける転送元または転送先のアドレスは4種類のアドレッシングモードで示されるようになっている。ここで、図5の演算内容の欄における記号「@」は、これに続く値をアドレスとする領域がデータの転送元または転送先であることを示し、また、記号「(,)」は、「,」の両側の値の和を示す。

【0025】すなわち、各転送命令のアドレッシングモードは、データの転送元または転送先が、(1)「@A、」である場合には、アドレスレジスタ間接アドレッシング、(2)同、「@(A、,d8)」である場合には、命令コードの「1001D、A、」または「1101D、A、」に続く8ビットのディスプレースメントd8を用いる8ビットディスプレースメント付きアドレスレジスタ間接アドレッシング、(3)同、「@(A、,D0)」である場合には、アドレスレジスタA、(ただしA、≠11)とデータレジスタD0(D0に固定)とを用いるインデックスアドレッシング、(4)同、「@abs8」である場合には、命令コードに続く8ビットの絶対アドレスabs8を用いる8ビット絶対アドレッシング、であることを表わしている。

【0026】なお、上記(3)のインデックスアドレッシングにおいては、アドレスレジスタをA0~A2に限定し、A3を除外しているが、これは、この種のアドレッシングは一般にストリング列や配列などのデータを転送する際に用いられ、そのようなデータは、通常、メモリのスタック領域を対象として転送されることが少なく、それゆえ、前述のようにアドレスレジスタA3がス

6

タックポインタ用として用いられる場合に、これをインデックスアドレッシングに用いることは希であるからである。

【0027】そして、上記(4)の8ビット絶対アドレッシングでは、インデックスアドレッシングで除外されたアドレスレジスタA3を示すレジスタ指定コードA、=11をダミーとして用い、転送元または転送先のデータレジスタをD0に限定した命令コードを割り当てている。ここで、8ビット絶対アドレッシングでデータ転送が行われる場合、データ処理装置からは、16ビットのアドレスのうち上位8ビットとして、例えば「00000000」など、あらかじめ設定された値が出力されるようになっている。

【0028】命令タイプ[R]は、1つのデータレジスタを対象とする単項演算命令で、命令コード「1011***D、,」または「1111***D、,」で表わされ、2ビットの「**」の値に応じて、算術シフト命令(右、左)、回転命令(右、左)、論理反転命令、値1の加減算命令、またはプログラム割り込み命令の8種類の演算命令であることを示している。

【0029】また、以上の説明で演算の割り当てられていない命令コードは、上位ビットが「1010」または「1110」であるものが各3つずつあるが、これらの命令コードは、アドレスレジスタA0~A3へのアドレス転送等、図示しない他の命令の割り当てや、16ビット以上の命令コードに拡張するためなどに用いられる。すなわち、使用頻度が少なく、プログラムサイズに与える影響が小さい命令などは、このように拡張命令コードとして設定すればよい。なお、このような命令コードの拡張については、本発明とは直接関係しないので詳細な説明を省略する。

【0030】上記のように構成されたデータ処理装置に最上位ビットが0である命令コード、例えば「01110001」が入力されると、レジスタ指定コード一致検出部13は下位2ビットの値とこれよりも上位の2ビットの値と、すなわち2つのレジスタ指定コードD、(=00)と、D、' (=01)とを比較し、一致していないことを示す信号を命令解読器14に出力する。

【0031】そこで、命令解読器14はデータレジスタD0、D1に制御信号を出力し、これらのデータレジスタD0、D1に保持されているデータを内部データバス17を介して演算器16に出力させる。命令解読器14は、また、命令コードの上位2ビットめから4ビットめの値「111」に基づいて論理和演算を行わせるための制御信号を演算器16に出力し、演算器16はデータレジスタD0、D1から出力されたデータの論理和演算を行う。演算結果は内部データバス17を介してデータレジスタD0に格納される。

【0032】一方、データ処理装置に「01110000」およびこれに続く8ビットの即値が入力されると、

50

レジスタ指定コード一致検出部13はD₀ = D₁ = 0であることを検出する。そこで、命令解読器14の制御により、データレジスタD₀に保持されているデータ、および命令バッファ11に蓄積されている8ビットの即値が内部データバス17を介して演算器16に入力され、論理和演算が行われて演算結果がデータレジスタD₀に格納される。

【0033】すなわち、従来のデータ処理装置のように、2つのレジスタ指定コードが互いに等しい場合でも等しくない場合と同じ演算を行わせるとすると、例えばデータレジスタD₀のデータどうしの論理和演算は実際には何もしていないことになる。これに対し、本実施例のデータ処理装置では、このような場合に異なる演算を行わせるようにすることにより、図5に「***」で表わされる3ビットのフィールドで16種類の命令を割り当てることができ、したがって、基本命令語長が8ビットであっても多くの種類の演算を実行させることが可能になっている。なお、例えば、減算命令についてはレジスタ指定コードが等しい場合でも減算を実行させるようにしてレジスタのクリア命令として用いられるようにするなど、一部の命令については同じ演算を行わせるようにしてもよい。

【0034】また、上記の例ではレジスタアレイ15内の8本のレジスタをデータレジスタD₀～D₃とアドレスレジスタA₀～A₃との、機能を特化した2組に分けることにより、2ビットのフィールドで1つのレジスタを指定することができる。それゆえ、8ビットの命令コードにソースレジスタとディスティネーションレジスタとの2つのレジスタ指定コードを含ませる場合でも、残りの4ビットで演算の種類を指定できるので、一層、短い基本命令語長で多くの種類の演算を実行させることができる。

【0035】このように、短い基本命令語長で多くの種類の演算を実行させることができるので、頻繁に用いられる基本的な演算（上記の例では、8種類のデータレジスタ間演算、8種類のデータレジスタ、即値間演算、8種類のデータレジスタ、メモリ間転送、および8種類の、1つのデータレジスタに対する単一オペランド演算）を短い命令語長の命令で実行させることができる。

【0036】したがって、上記のようなレジスタおよび演算命令の構成によって、例えばC言語などの高級言語による処理を効果的に実行させるための必要最小限の環*

*境を提供することができ、しかもプログラムサイズを小さく抑え得るとともに処理速度の高速化を図ることができる。なお、上記実施例においては、命令コードにおける即値や絶対アドレスは8ビットで、命令に応じて16ビットに拡張される例を示したが、これに限らず必要に応じて16ビットの即値を用い得るようにしてもよい。

【0037】さらに、レジスタ構成や命令コードの構造、演算の種類、オペレーションコードのビット割り当て等は上記のものに限らず、データ処理装置の用途に応じた種々の構成等においても本発明を適用することが可能である。また、レジスタ指定コード一致検出部13を設ける構成に限らず、例えば命令解読器14が、レジスタ指定コードが一致する場合と一致しない場合とで異なる命令として解読するように構成してもよい。

【0038】

【発明の効果】以上説明したように、本発明によれば、2つのレジスタ指定コードが互いに異なる場合と同一である場合とで、異なる処理を行わせるインストラクション実行制御部を備えていることにより、同一のオペレーションコードに対して2通りの処理を行わせることができるので、短い命令語長で多くの種類の演算を実行させることができ、したがってプログラムサイズを小さく抑えることができるという効果を奏する。

【図面の簡単な説明】

【図1】データ処理装置の要部の構成を示すブロック図である。

【図2】レジスタアレイの詳細な構成を示す説明図である。

【図3】状態レジスタPSWの詳細な構成を示す説明図である。

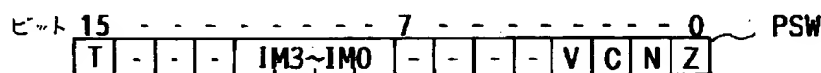
【図4】命令コードの構造の例を示す説明図である。

【図5】命令コードの割り当ての例を示す説明図である。

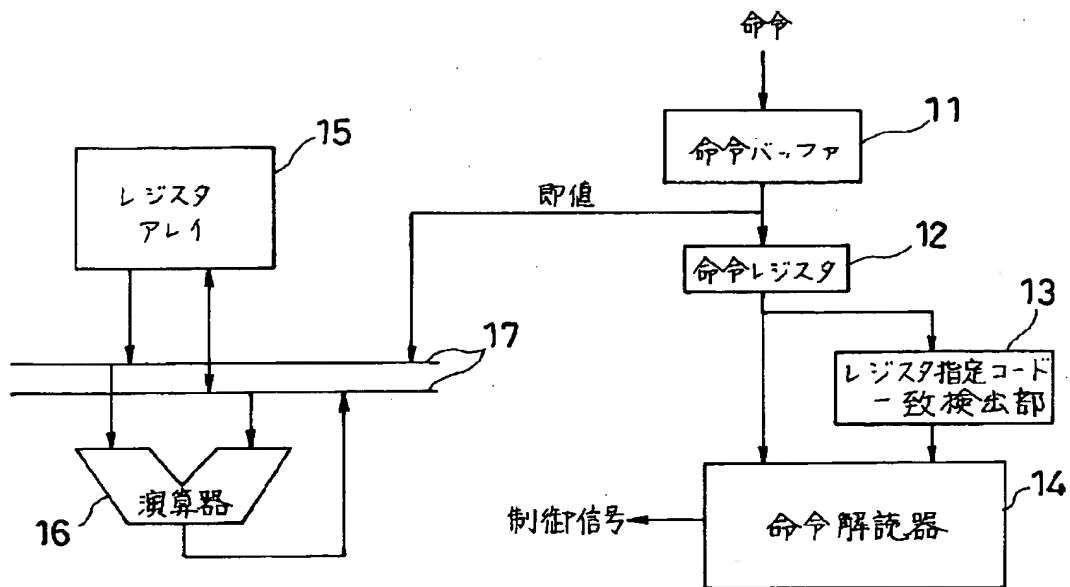
【符号の説明】

- 11 命令バッファ
- 12 命令レジスタ
- 13 レジスタ指定コード一致検出部
- 14 命令解読器
- 15 レジスタアレイ
- 16 演算器
- 17 内部データバス

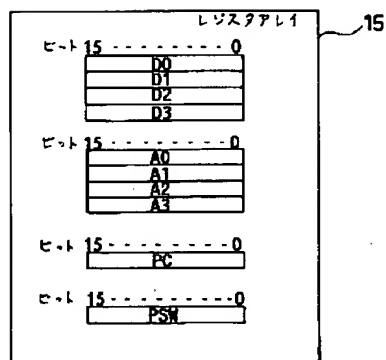
【図3】



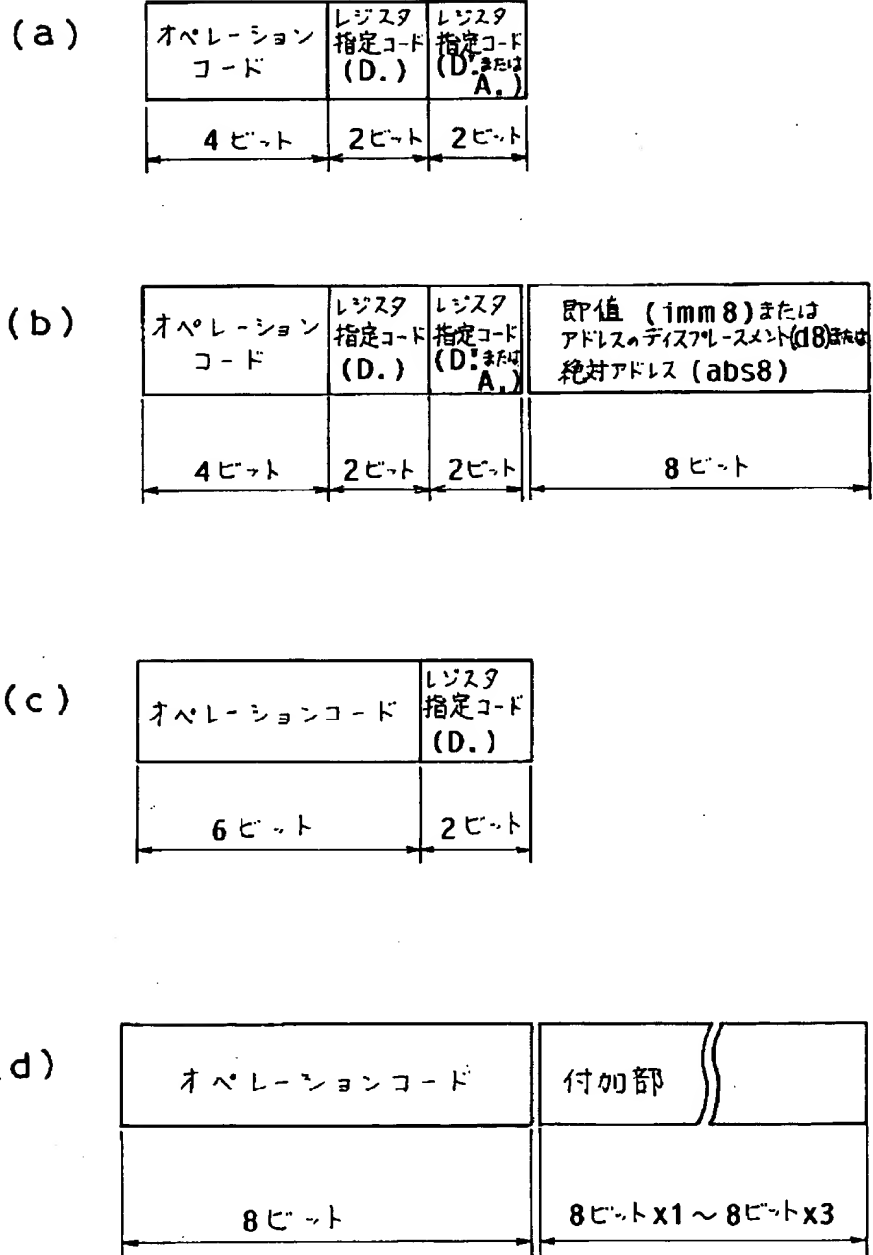
【図1】



【図2】



【図4】



【図5】

命令コード	命令コードの構造	命令のタイプ	演算内容
0***D.D! (ただしD.≠D.')	(a)	[R-R]	***の値に応じて MOV (転送) ADD (キャリ無し加算) SUB (ボロー無し加算) CMP (比較) ADDC (キャリ付き加算) SUBC (ボロー付き加算) AND (論理積) OR (論理和)
0***D.D'+imm8 (ただしD.=D.')	(b)	[R-imm8]	同上
1000D.A.	(a)	[LD]	転送元 @A. (アドレスレジスタA.の値を アドレスとする領域)
1001D.A.+d8	(b)		@(A.,d8) (アドレスレジスタA.の値+d8を アドレスとする領域)
1010D.A. (ただしA.≠11)	(a)		@(A.,D0) (アドレスレジスタA.の値 +データレジスタD0の値を アドレスとする領域)
1010D.A.+abs8 (ただしD.=00, A.=11)	(b)		@abs8 (abs8を アドレスとする領域)
1100D.A.	(a)	[ST]	転送先 @A. (アドレスレジスタA.の値を アドレスとする領域)
1101D.A.+d8	(b)		@(A.,d8) (アドレスレジスタA.の値+d8を アドレスとする領域)
1110D.A. (ただしA.≠11)	(a)		@(A.D0) (アドレスレジスタA.の値 +データレジスタD0の値を アドレスとする領域)
1110D.A.+abs8 (ただしD.=00, A.=11)	(b)		@abs8 (abs8を アドレスとする領域)
1011**D.	(c)	[R]	**の値に応じて ASR (算術シフト右) ASL (算術シフト左) ROR (回転右) ROL (回転左)
1111**D.	(c)		**の値に応じて NOT (論理反転) INC (値1を加算) DEC (値1を減算) PI (プログラム割り込み)